

UNIVERSIDADE DO ESTADO DE MATO GROSSO

MARIA DA CONCEIÇÃO BRAGA DA SILVA

EXERCÍCIO P2 - ESTIMULADOR MAGNÉTICO

SINOP - 2025

Relatório Técnico – Simulação do Circuito RLC com Diodo - Estimulador magnético

Análise Temporal do Circuito RLC com Diodo

Aluna: Maria da Conceição Braga da Silva

Disciplina: Eletrônica de Potência

Objetivo

O objetivo deste trabalho é analisar, através de simulação numérica, o comportamento temporal da corrente no indutor e da tensão no capacitor em um circuito RLC que apresenta duas etapas distintas de funcionamento, de acordo com o estado de condução de um diodo, a partir da metodologia de Estados assumidos.

A simulação foi realizada em Python, utilizando bibliotecas científicas para modelagem matemática e visualização dos resultados.

Fundamentos

O circuito estudado é composto por um indutor (L), um capacitor (C), um resistor (R) e um diodo. Ele passa por dois regimes de operação:

Etapa 1 ou Estado 1 - Diodo Desligado (LC em série):

- O capacitor está inicialmente carregado com uma tensão (900 V);
- Ao iniciar a descarga, a corrente flui pelo indutor e há troca de energia entre L e C ;
- O circuito obedece às equações diferenciais:

```
di_dt = vC / L                      # tensão do capacitor acelera a corrente
dv_dt = -i / C                       # corrente descarrega o capacitor
```

Etapa 2 ou Estado 2 - Diodo Conduz (R, L e C em paralelo):

- Quando a tensão do capacitor tende a inverter (ficar negativa), o diodo entra em condução e o circuito se rearranja em paralelo entre os elementos R , L e C ;
- As equações passam a ser:

```
di_dt = (vC - R * i) / L
dv_dt = -(i + vC / R) / C
```

* Essas equações foram resolvidas numericamente para se obter a resposta temporal.

Metodologia de Simulação

A simulação foi desenvolvida em Python, utilizando os seguintes módulos:

- numpy: manipulação de vetores e operações matemáticas;
- matplotlib.pyplot: geração dos gráficos;
- scipy.integrate.solve_ivp: integração numérica das equações diferenciais.

Definição dos parâmetros do circuito:

```
# Parâmetros do circuito
V0 = 900           # [V] tensão inicial do capacitor (vC)
L = 11e-6          # [H] indutância da bobina
R = 85e-3          # [Ohm] resistência
C = 180e-6         # [F] capacidade
```

*Esses valores representam um circuito do Estimulador magnético trazido no exercício, trata-se de uma descarga controlada, com resistência pequena (baixa perda) e indutância moderada.

Modelagem matemática

A função circuito(t, y) descreve o comportamento dinâmico das variáveis i (corrente no indutor) e vC (tensão no capacitor):

```
def circuito(t, y):
    i, vC = y
    # Etapa 1: diodo desligado (C e L em série)
    if vC > 0:
        di_dt = vC / L      # tensão do capacitor acelera a corrente
        dv_dt = -i / C      # corrente descarrega o capacitor
    # Etapa 2: diodo conduz (C, R e L em paralelo)
    else:
        di_dt = (vC - R * i) / L
        dv_dt = -(i + vC / R) / C
    return [di_dt, dv_dt]
```

O condicional if representa a mudança de configuração quando o diodo conduz.

- Enquanto $vC > 0$, o diodo está bloqueado;
- quando $vC \leq 0$, ele entra em condução e altera a tipologia do circuito.

Condições iniciais e intervalo de tempo:

```
# Condições iniciais e tempo de simulação
i0 = 0.0
v0 = v0
condicoes_iniciais = [i0, v0]

t0 = 0.0
tf = 600e-6
t_eval = numpy.linspace(t0, tf, 5000)
```

*O capacitor começa carregado com 900 V e o indutor sem corrente. A simulação vai de 0 a 600 μ s, com 5.000 pontos igualmente espaçados (t_eval) — o que garante alta resolução temporal e gráficos suaves.

Integração numérica

A função `solve_ivp` resolve as equações diferenciais no intervalo de tempo definido:

```
# Resolução numérica
solucao = solve_ivp(circuito, [t0, tf], condicoes_iniciais, t_eval =
t_eval)
tempo = solucao.t
```

Ela retorna dois vetores principais:

```
corrente = solucao.y[0]
tensao = solucao.y[1]
```

Apresentação dos resultados

Os resultados são apresentados em dois gráficos empilhados:

```
# Gráficos
matplotlib.pyplot.figure(figsize=(9, 6))
# Corrente no indutor
matplotlib.pyplot.subplot(2, 1, 1) #Gráfico 1
matplotlib.pyplot.plot(tempo * 1e6, corrente, color='blue')
matplotlib.pyplot.title("Resposta Temporal do Circuito RLC com Diodo - INDUTOR e CAPACITOR")
matplotlib.pyplot.ylabel("Corrente no Indutor [A]")
matplotlib.pyplot.grid(True)
# Tensão no capacitor
matplotlib.pyplot.subplot(2, 1, 2) #Gráfico 2
matplotlib.pyplot.plot(tempo * 1e6, tensao, color='green')
```

```

matplotlib.pyplot.xlabel("Tempo [μs]")
matplotlib.pyplot.ylabel("Tensão no Capacitor [V]")
matplotlib.pyplot.grid(True)

matplotlib.pyplot.tight_layout()
matplotlib.pyplot.show()

```

O comando subplot(2,1,1) cria o primeiro gráfico (corrente) e subplot(2,1,2) o segundo (tensão), ambos com o mesmo eixo de tempo.

O eixo X é multiplicado por 1e6 (μs) para converter de segundos para microssegundos, tornando a escala mais intuitiva.

Resultados Obtidos

A simulação resultou em (no Terminal):

```

(venv) PS
C:\Users\User\onedrive\documentos\EletrônicadePotência\EPmaria> &
C:/Users/User/OneDrive/Documentos/EletrônicadePotência/EPmaria/ven
v/Scripts/python.exe
c:/Users/User/OneDrive/Documentos/EletrônicadePotência/EPmaria/mai
n.py          otência/EPmaria/venv/Scr

===== Resultados da Simulação =====
Corrente máxima no indutor: 3648.9243 A
=====

```

Análise do comportamento

A corrente no indutor cresce rapidamente devido à alta tensão inicial do capacitor e à baixa resistência da malha. Após atingir o pico, a corrente decai exponencialmente enquanto o indutor libera energia no resistor. A tensão no capacitor cai de 900 V até valores próximos de zero. A pequena tensão negativa observada ocorre porque o modelo matemático permite uma leve inversão de polaridade - algo que, na prática, seria bloqueado pelo diodo ideal.

Considerações sobre o Diodo

Durante o funcionamento:

- Antes da condução: o diodo atua como um circuito aberto, e o capacitor transfere energia para o indutor.
- Após a condução: o diodo passa a agir como um curto, conectando o resistor em paralelo ao indutor e capacitor, dissipando energia até o circuito atingir o equilíbrio.

A presença do diodo garante que a tensão não inverta de forma significativa, protegendo o capacitor e controlando o sentido da corrente.

Conclusões

O modelo desenvolvido em Python permitiu visualizar o comportamento dinâmico de um circuito RLC com diodo, comparar as etapas de funcionamento (diodo bloqueado e conduzindo) e também observar a forma de onda realista de corrente e tensão.

A simulação confirma os conceitos teóricos estudados em Eletrônica de Potência: o indutor se opõe a variações bruscas de corrente, o capacitor armazena e libera energia, e o diodo define a direção de condução no circuito.

Código completo apresentado

```
# Exercício RLC - Autor: Maria Braga - Disciplina: Eletrônica de Potência - P2

# Objetivo:
# Obter a resposta temporal da corrente no indutor e da
# tensão no capacitor durante o funcionamento do circuito.

# Etapa 1: circuito série (Capacitor + Indutor)
# Etapa 2: circuito paralelo (Capacitor + Resistor + Indutor)

import numpy
import matplotlib.pyplot
from scipy.integrate import solve_ivp

# Parâmetros do circuito
V0 = 900          # [V] tensão inicial do capacitor (Vc)
L = 11e-6         # [H] indutância da bobina
R = 85e-3         # [Ohm] resistência
C = 180e-6        # [F] capacidade

# Equações diferenciais do circuito

def circuito(t, y):
    i, vC = y

    # Etapa 1: diodo desligado (C e L em série)
    if vC > 0:
        di_dt = vC / L           # tensão do capacitor acelera a corrente
        dv_dt = -i / C           # corrente descarrega o capacitor

    # Etapa 2: diodo conduz (C, R e L em paralelo)
    else:
        di_dt = (vC - R * i) / L
        dv_dt = -(i + vC / R) / C

    return [di_dt, dv_dt]

# Condições iniciais e tempo de simulação

i0 = 0.0
v0 = V0
condicoes_iniciais = [i0, v0]

t0 = 0.0
tf = 600e-6
t_eval = numpy.linspace(t0, tf, 5000)

# Resolução numérica

solucao = solve_ivp(circuito, [t0, tf], condicoes_iniciais, t_eval = t_eval)
```

```

tempo = solucao.t
corrente = solucao.y[0]
tensao = solucao.y[1]

# Gráficos

matplotlib.pyplot.figure(figsize=(9, 6))

# Corrente no indutor
matplotlib.pyplot.subplot(2, 1, 1) #Gráfico 1
matplotlib.pyplot.plot(tempo * 1e6, corrente, color='blue')
matplotlib.pyplot.title("Resposta Temporal do Circuito RLC com Diodo - INDUTOR e CAPACITOR")
matplotlib.pyplot.ylabel("Corrente no Indutor [A]")
matplotlib.pyplot.grid(True)

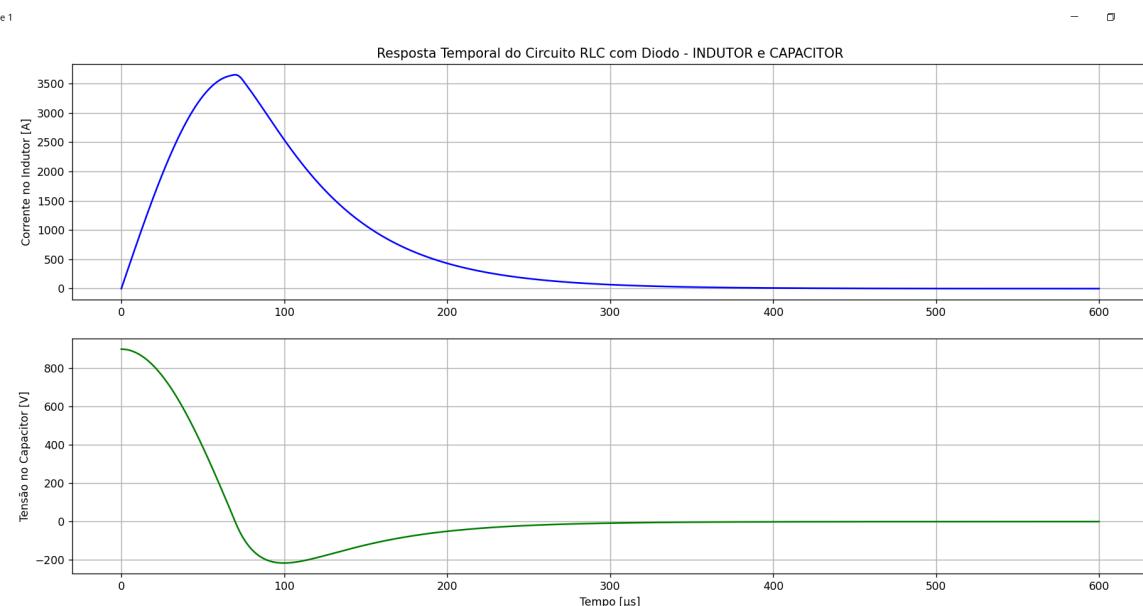
# Tensão no capacitor
matplotlib.pyplot.subplot(2, 1, 2) #Gráfico 2
matplotlib.pyplot.plot(tempo * 1e6, tensao, color='green')
matplotlib.pyplot.xlabel("Tempo [μs]")
matplotlib.pyplot.ylabel("Tensão no Capacitor [V]")
matplotlib.pyplot.grid(True)

matplotlib.pyplot.tight_layout()
matplotlib.pyplot.show()

# Resultados principais
print("===== Resultados da Simulação =====")
print(f"Corrente máxima no indutor: {numpy.max(corrente):.4f} A")
print(f"Tensão mínima no capacitor: {numpy.min(tensao):.4f} V")
print("=====")

```

Resultado



FIM!!!